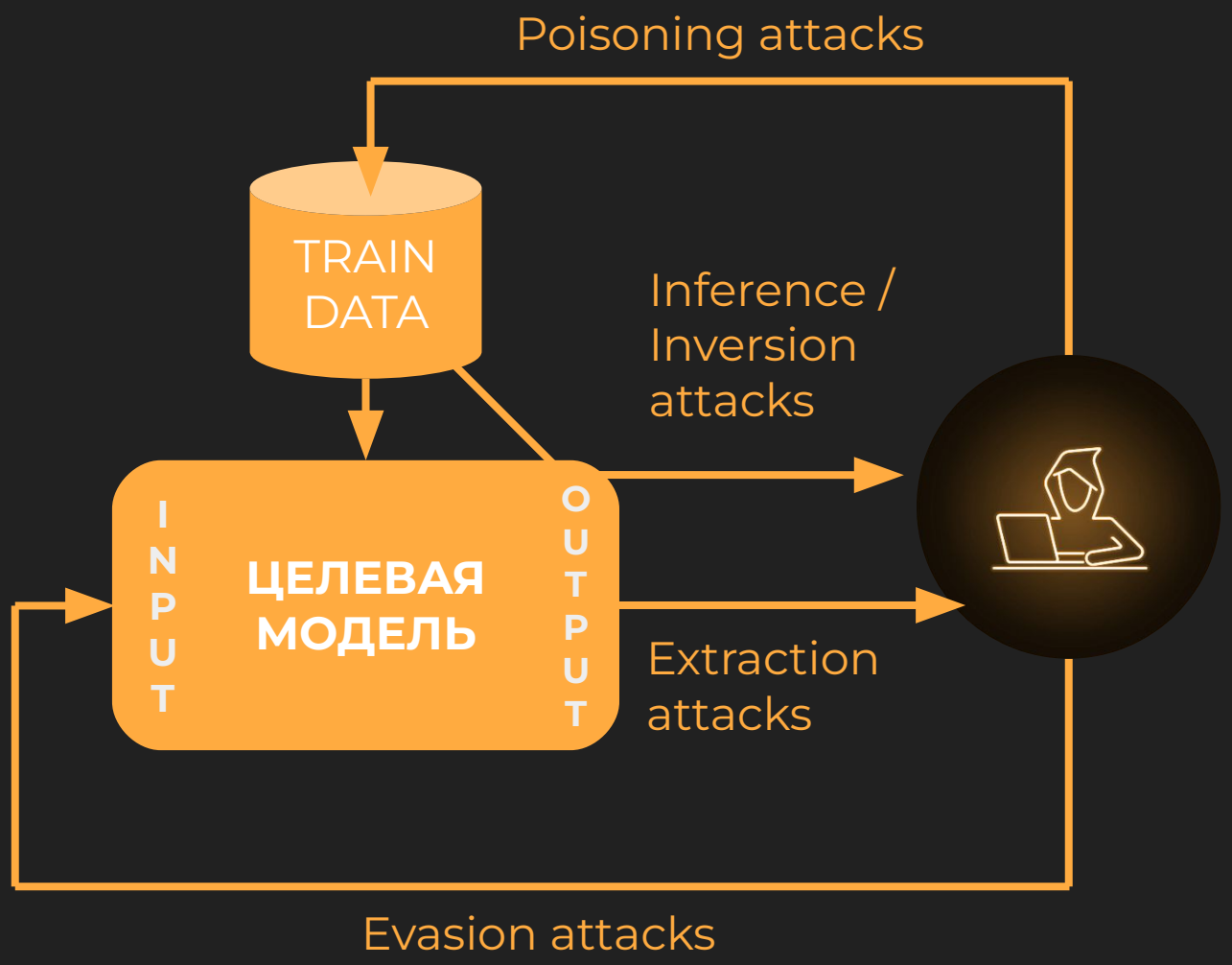
ИИ глазами хакера

Елизавета Тишина

Специалист по анализу защищённости, DeteAct

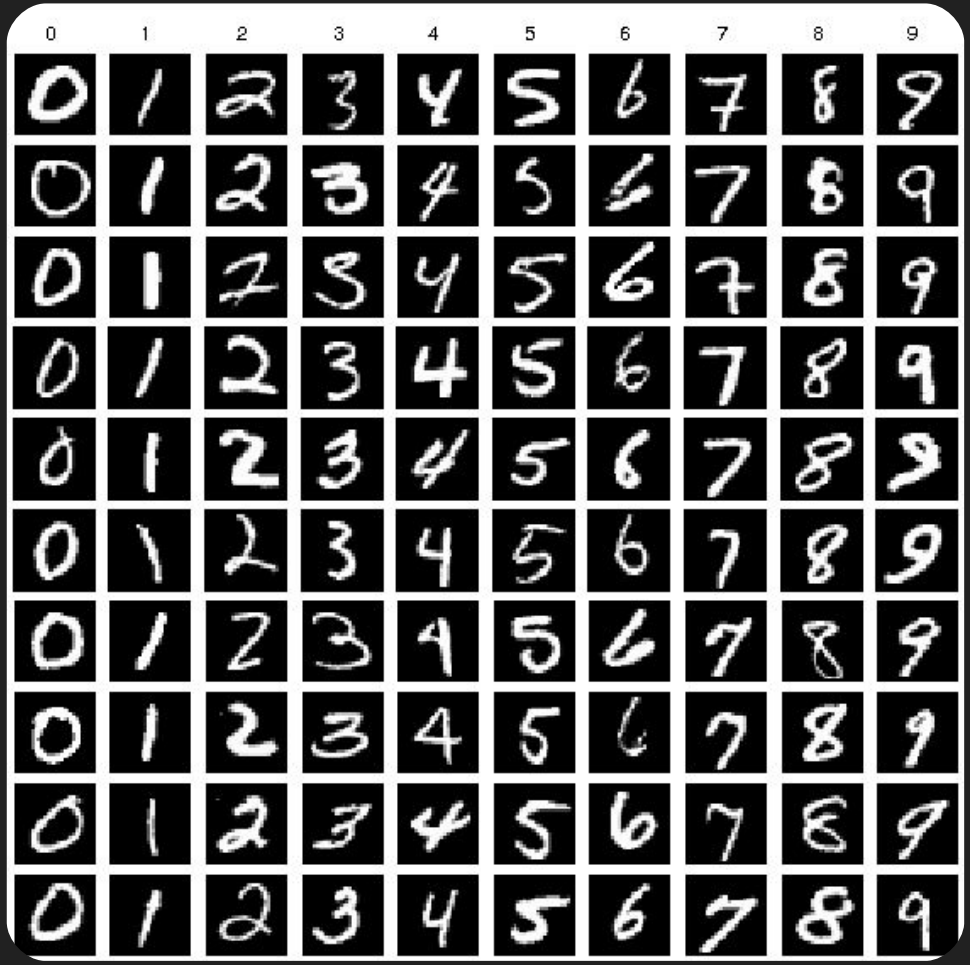
29 Октября, 2022

Потенциальные угрозы

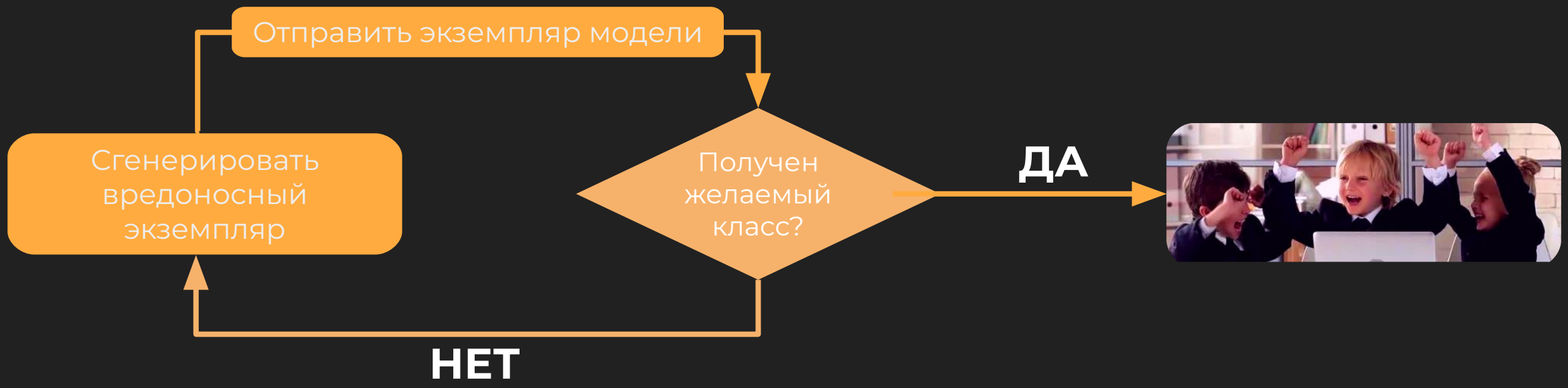


Обзор модели-примера

- **Тренировочные данные**: MNIST
- **Условия**: API доступ к модели
- **Знания атакующего**: –

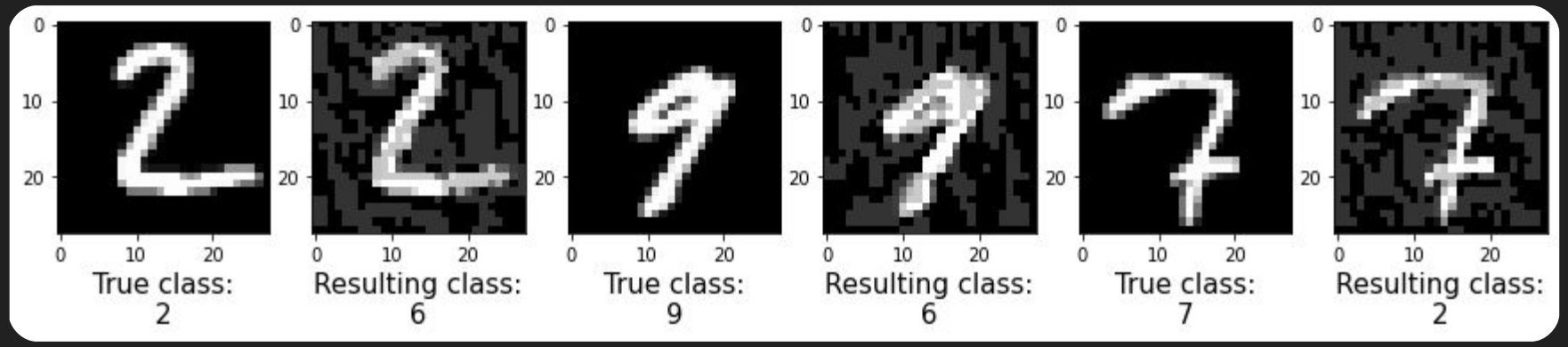


Атаки обхода: Обзор



Атаки обхода: Пример

github.com/qwqoro/ML-Talk



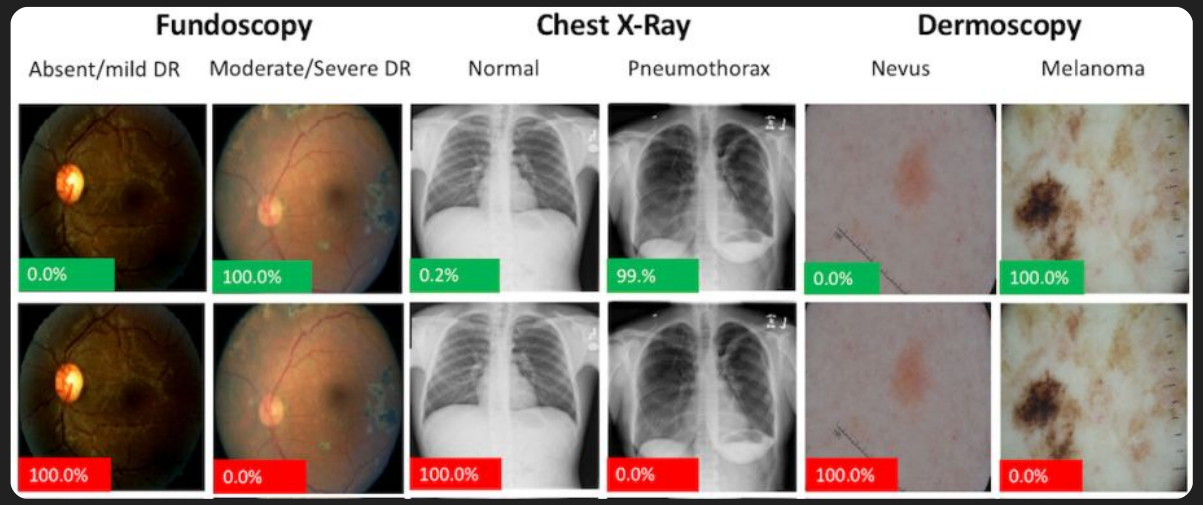
Подход: Fast Gradient Method [arXiv:1412.6572]

Атаки обхода: Влияние

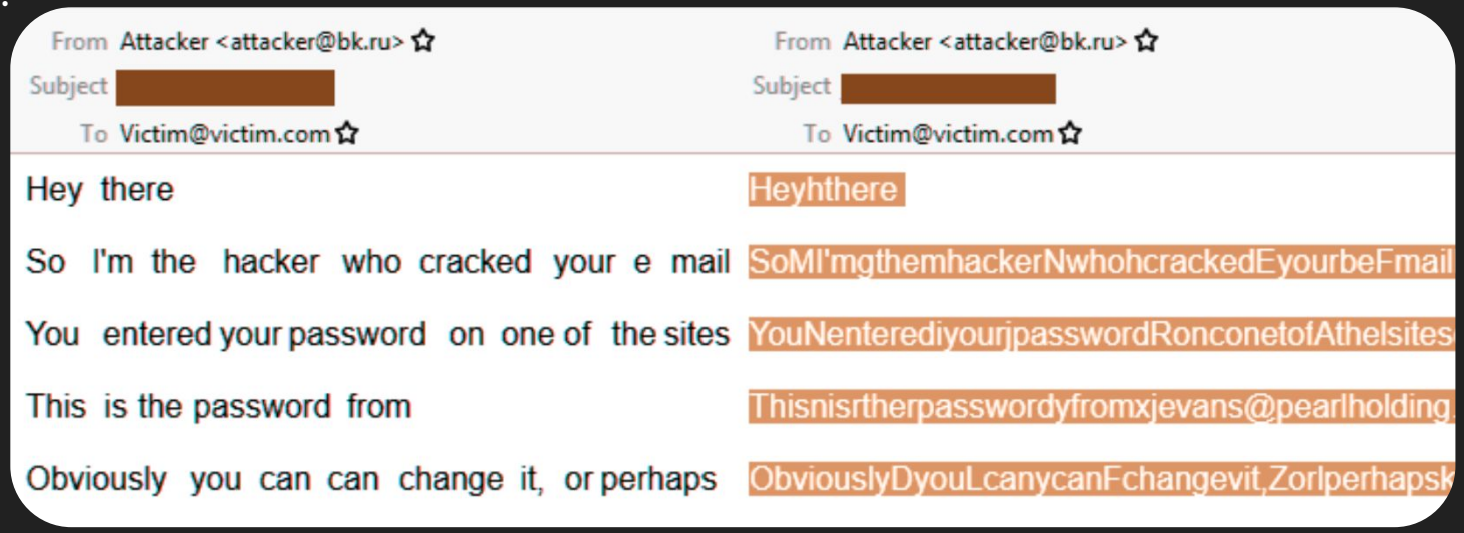
- Спуфинг систем верификации

- Обход фильтров:
  - антивирусов
  - спам-фильтров
  - фильтров рекламы
  - . . .

- Угрожающие жизни ситуации:
  - неверные диагнозы
  - неверные решения автопилота
  - . . .



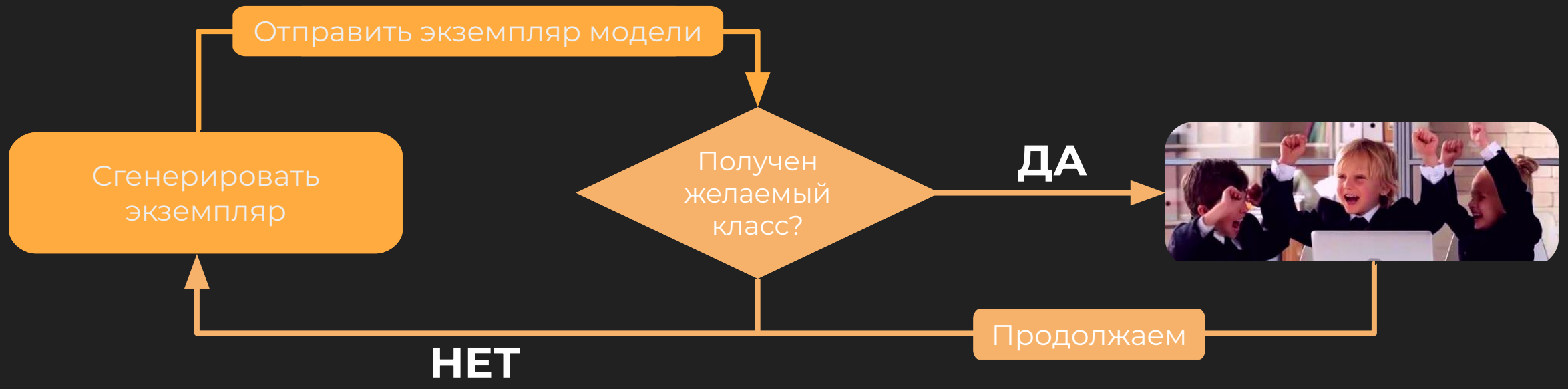
[arXiv:1804.05296]



Атаки обхода: Защита

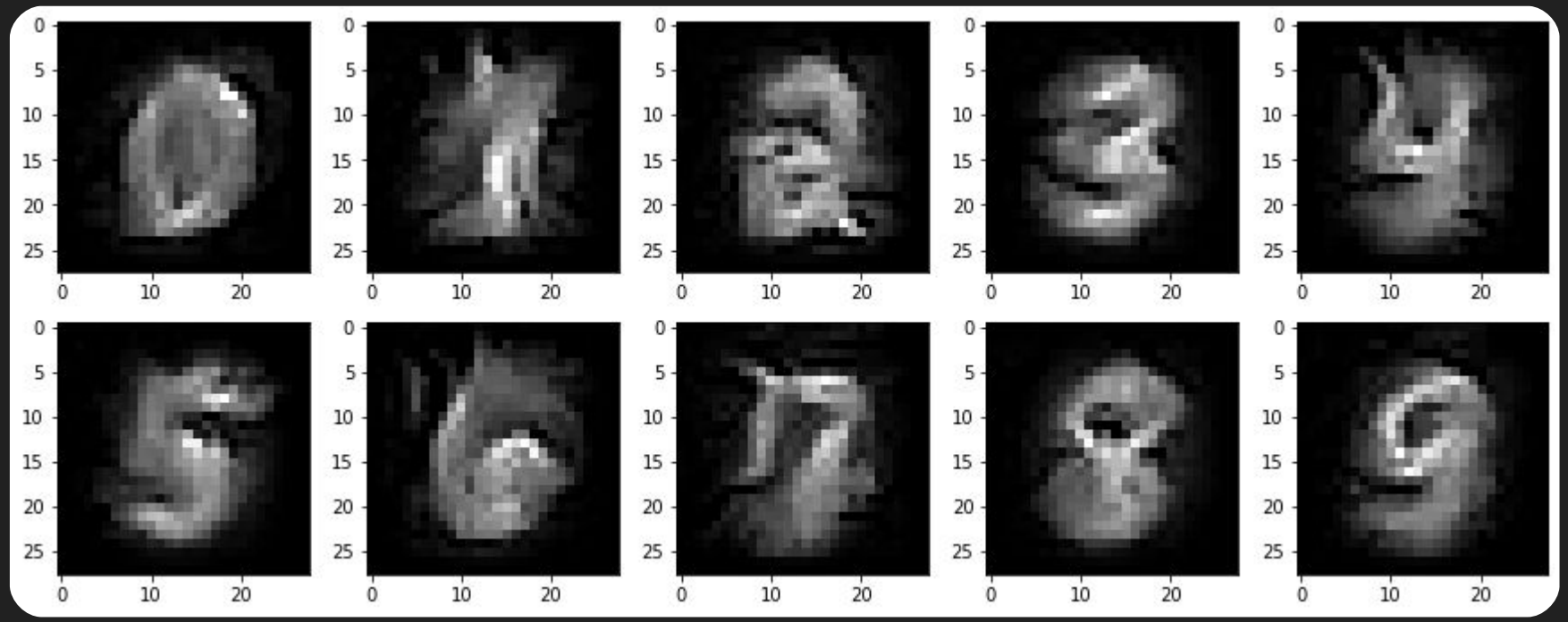
- Человеческий контроль
- Ограничение числа запросов к модели
- Detector:
  Дополнительная модель для оценки вредоносности экземпляров
- Adversarial training:
  Обучение модели на вредоносных экземплярах
- Defensive distillation

Атаки выворачивания модели: Обзор



Атаки выворачивания модели: Пример

github.com/qwqoro/ML-Talk

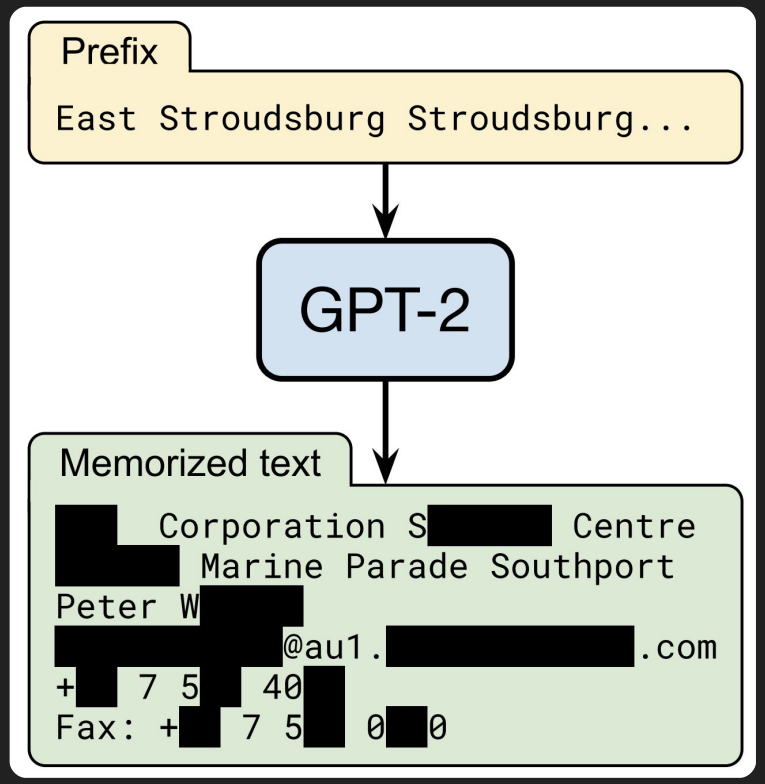


Подход: MIFace [DOI:10.1145/2810103.2813677]

Атаки выворачивания модели: Влияние

Утечка данных:

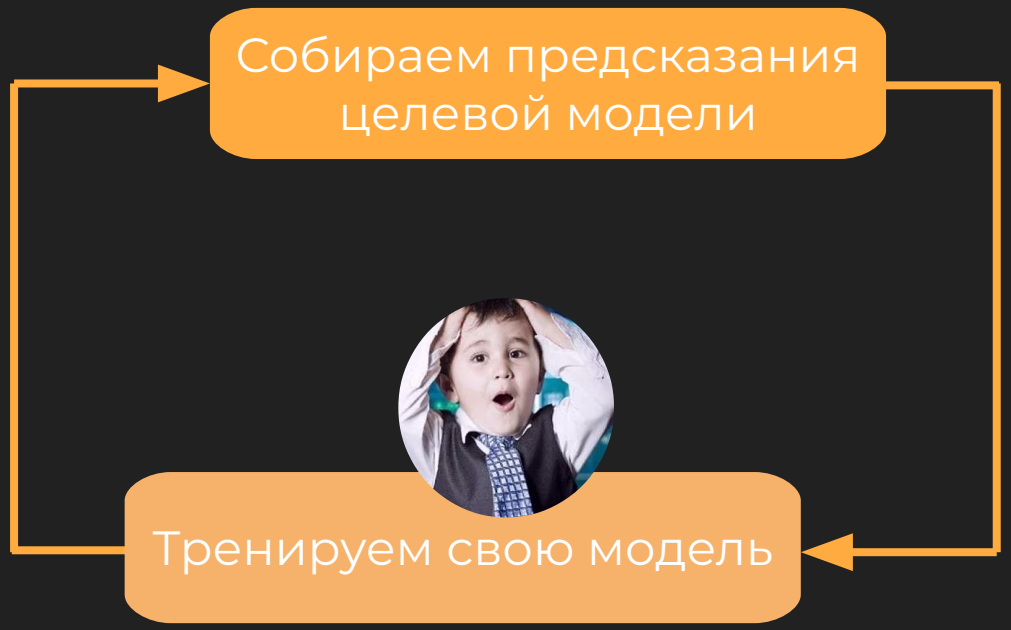
- Содержимое документов
- Медицинские записи
- Пароли
- PIN коды
- ...



[arXiv:2012.07805]

Атаки извлечения модели: Обзор & Влияние

- Нарушение права интеллектуальной собственности



Атаки извлечения / выворачивания модели: Защита

- ...
- Model retraining
- Post-processing
- Выбор случайной модели из нескольких различных моделей

Adversarial Robustness Toolbox

github.com/Trusted-AI/adversarial-robustness-toolbox

- Attacks
- Defences
- Estimators
- Evaluations
- Metrics
- Data generators

```
from art.attacks.inference.model_inversion import MIFace

x_average = np.zeros((10, 28, 28, 1)) + np.mean(x_test, axis=0)

attackInversion = MIFace(classifier, max_iter=25000, threshold=1.0, batch_size=10, window_length=128)
inverted = attackInversion.infer(x_average, y=np.arange(10))
```

```
Model inversion: 100%|                    | 10/10 [10:13<00:00, 61.38s/it]
```

```
from art.attacks.evasion import FastGradientMethod

# Generation of adversarial examples
attackEvasion = FastGradientMethod(estimator=classifier, eps=0.2, batch_size=64)
x_adv = attackEvasion.generate(x_test)

# Predicting and evaluating accuracies of predictions on both initial data samples and adversarial ones
predictions = (classifier.predict(x_test), classifier.predict(x_adv))
accuracies = (np.sum(np.argmax(predictions[0], axis=1) == np.argmax(y_test, axis=1)) / len(y_test),
              np.sum(np.argmax(predictions[1], axis=1) == np.argmax(y_test, axis=1)) / len(y_test))

print(f"Accuracy of predictions (initial data): {accuracies[0] * 100} %")
print(f"Accuracy of predictions  (adversarial): {accuracies[1] * 100} %")
```

```
Accuracy of predictions (initial data): 98.16 %
Accuracy of predictions  (adversarial): 41.88 %
```

[ART] Атаки извлечения модели: Пример атаки

github.com/qwqoro/ML-Talk

```
from art.attacks.extraction import CopycatCNN

# Training a substitute model based on the target model
attackExtraction = CopycatCNN(classifier, batch_size_fit=10, batch_size_query=10, nb_epochs=10, nb_stolen=100)
extracted = attackExtraction.extract(x_test, thieved_classifier=res)
```

```
Train on 100 samples
Epoch 1/10
100/100 [==============================] - 0s 2ms/sample - loss: 2.2616 - accuracy: 0.1500
Epoch 2/10
100/100 [==============================] - 0s 740us/sample - loss: 2.0509 - accuracy: 0.2600
Epoch 3/10
100/100 [==============================] - 0s 658us/sample - loss: 1.7601 - accuracy: 0.4600
Epoch 4/10
100/100 [==============================] - 0s 658us/sample - loss: 1.4344 - accuracy: 0.5800
Epoch 5/10
100/100 [==============================] - 0s 702us/sample - loss: 1.1608 - accuracy: 0.6700
Epoch 6/10
100/100 [==============================] - 0s 707us/sample - loss: 0.9168 - accuracy: 0.7400
Epoch 7/10
100/100 [==============================] - 0s 804us/sample - loss: 0.7963 - accuracy: 0.7500
Epoch 8/10
100/100 [==============================] - 0s 700us/sample - loss: 0.6875 - accuracy: 0.7900
Epoch 9/10
100/100 [==============================] - 0s 570us/sample - loss: 0.6211 - accuracy: 0.8100
Epoch 10/10
100/100 [==============================] - 0s 619us/sample - loss: 0.5331 - accuracy: 0.8100
```

```
# Making predictions with use of both original and extracted versions of the target model and evaluating their similarity
victim_predictions = np.argmax(model.predict(x_test), axis=1)
thieved_predictions = np.argmax(extracted.predict(x_test), axis=1)
accuracy = np.sum(victim_predictions == thieved_predictions) / len(victim_predictions)

print(f"Similarity of predictions: {accuracy * 100} %")
```

```
Similarity of predictions: 69.31 %
```

Подход: Copycat CNN [arXiv:1806.05476]

Counterfit

github.com/Azure/counterfit

```
/content/counterfit# python counterfit.py

                                  __                 ____ __
    ________  __  ______ / /____  ____/ __(_) /_
   / ___/ __ \/ / / / __ \/ __/ _ \/ ___/ /_/ / __/
  / /__/ /_/ / /_/ / /_/ / /_/  __/ /  / / / /
  \___/\____/\__,_/_/ /_/\__/\___/_/  /_/ /_/\__/

                  Version: 1.0.0


counterfit> list targets
```

Name	Model Type	Data Type	Input Shape	# Samples	Endpoint	Loaded
creditfraud	BlackBox	tabular	(30,)	(not loaded)	creditfraud_sklearn_pipeline.pkl	False
digits_blackbox	BlackBox	image	(1, 28, 28)	(not loaded)	mnist_sklearn_pipeline.pkl	False
digits_keras	keras	image	(28, 28, 1)	(not loaded)	mnist_model.h5	False
movie_reviews	BlackBox	text	(1,)	(not loaded)	movie_reviews_sentiment_analysis.pt	False
satellite	BlackBox	image	(3, 256, 256)	(not loaded)	satellite-image-params-airplane-stadium.h5	False

```
counterfit> list frameworks
```

Framework	# Attacks
art	(not loaded)
augly	(not loaded)
textattack	(not loaded)

Counterfit

github.com/Azure/counterfit

```
counterfit> list attacks
```

Name	Category	Type	Tags	Framework
A2TYoo2021	BlackBox	EvasionAttack	text	textattack
BAEGarg2019	BlackBox	EvasionAttack	text	textattack
BERTAttackLi2020	BlackBox	EvasionAttack	text	textattack
CLARE2020	BlackBox	EvasionAttack	text	
CheckList2020	BlackBox	EvasionAttack	text	
DeepWordBugGao2018	BlackBox	EvasionAttack	text	
FasterGeneticAlgorithmJia2019	BlackBox	EvasionAttack	text	
GeneticAlgorithmAlzantot2018	BlackBox	EvasionAttack	text	
HotFlipEbrahimi2017	BlackBox	EvasionAttack	text	
IGAWang2019	BlackBox	EvasionAttack	text	
InputReductionFeng2018	BlackBox	EvasionAttack	text	
Kuleshov2017	BlackBox	EvasionAttack	text	
MorpheusTan2020	BlackBox	EvasionAttack	text	
PSOZang2020	BlackBox	EvasionAttack	text	
PWWSRen2019	BlackBox	EvasionAttack	text	
Pruthi2019	BlackBox	EvasionAttack	text	
Seq2SickCheng2018BlackBox	BlackBox	IntegrityAttack	text	
TextBuggerLi2018	BlackBox	EvasionAttack	text	
TextFoolerJin2019	BlackBox	EvasionAttack	text	
BoundaryAttack	BlackBox	EvasionAttack	image	
CarliniL0Method	WhiteBox	EvasionAttack	image, tabular	art
CarliniLInfMethod	WhiteBox	EvasionAttack	image, tabular	art
CopycatCNN	BlackBox	ExtractionAttack	image	art
DeepFool	WhiteBox	EvasionAttack	image, tabular	art
ElasticNet	WhiteBox	EvasionAttack	image, tabular	art
FunctionallyEquivalentExtraction	BlackBox	ExtractionAttack	image, tabular	art
HopSkipJump	BlackBox	EvasionAttack	image, tabular	art
KnockoffNets	BlackBox	ExtractionAttack	image, tabular	art
LabelOnlyDecisionBoundary	WhiteBox	InferenceAttack	image, tabular	art
MIFace	WhiteBox	InferenceAttack	image, tabular	art
NewtonFool	WhiteBox	EvasionAttack	image, tabular	art
ProjectedGradientDescentCommon	WhiteBox	EvasionAttack	image, tabular	art
SaliencyMapMethod	WhiteBox	EvasionAttack	image, tabular	art
SimBA	WhiteBox	EvasionAttack	image	art
SpatialTransformation	WhiteBox	EvasionAttack	image, tabular	art
UniversalPerturbation	WhiteBox	EvasionAttack	image	art
VirtualAdversarialMethod	WhiteBox	EvasionAttack	image	art
Wasserstein	WhiteBox	EvasionAttack	image	art
Blur	BlackBox	CommonCorruption	image	augly
Brightness	BlackBox	CommonCorruption	image	augly
ChangeAspectRatio	BlackBox	CommonCorruption	image	augly
ClipImageSize	BlackBox	CommonCorruption	image	augly
ColorJitter	BlackBox	CommonCorruption	image	augly
Contrast	BlackBox	CommonCorruption	image	augly

[Counterfit] Атаки обхода: Пример

github.com/qwqoro/ML-Talk

```
digits_blackbox> use HopSkipJump

[+] New HopSkipJump (419f7593) created
[+] Using 419f7593

digits_blackbox>419f7593> set --sample_index 1 --max_eval 1500 --max_iter 10
```

```
digits_blackbox>419f7593> run

[-] Running attack HopSkipJump with id 419f7593 on digits_blackbox)

[-] Preparing attack...
[-] Running attack...
```

Success	Elapsed time	Total Queries
1/1	0.7	2390 (3504.1 query/sec)

Sample Index	Input Label (conf)	Adversari… Label (conf)	Max Abs Chg.	Adversarial Input
1	0 (1.0000)	6 (0.9809)	4.7776	counterfit/targets/digits_blackbox/results/419f7593/digits_blackbox-f03b8b22-f

```
[+] Attack completed 419f7593 (HopSkipJump)
```

Подход: HopSkipJump [arXiv:1904.02144]

[Counterfit] Атаки обхода: Пример

github.com/qwqoro/ML-Talk

Изначальное изображение	Вредоносная версия изображения	Разница (преувеличена)	Изнач. класс	Изначальная уверенность	Итоговый класс	Итоговая уверенность
			0	100%	6	98%
			1	100%	8	56%
			2	100%	4	73%

Может быть интересно 👀

- Коллаборация Google, OpenAI, Apple, Stanford, Berkeley и Northeastern University:
  Извлечение тренировочных данных из языковых моделей +
  демонстрация атаки на GPT-2
  - 🔗arXiv:2012.07805
- MIT:
  Демонстрация присутствия вредоносных 3D экземпляров в физическом мире +
  демонстрация атак обхода против нейросетей, разработанных Google
  - 🔗arXiv:1707.07397
  - 🔗arXiv:1804.08598

@qwqoro

qwqoro/ML-Talk

DETEACT